



8

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Pre-Amendment

8

T.P.

04/20/07

In re application of: Andrew R. Osborn

Serial No.: 09/982,601

Group Art Unit: 2662

Filed: October 18, 2001

Examiner: To Be Assigned

For: METHOD OF COMMUNICATING ACROSS AN
OPERATING SYSTEM

Attorney Docket No.: 65,270-005

PRELIMINARY AMENDMENT

RECEIVED

MAR 21 2003

Technology Center 2600

Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

Please amend the application as follows;

IN THE CLAIMS:

Claims 1-34 remain unchanged.

Please replace claim 35 with the following;

35. (Amended) A method of communicating across an operating system using a plurality of processes and a plurality of memory sources disposed within one or more processors, said method comprising the steps of:

detecting an event within the system;

extracting an initial process address from one of the memory sources to

determine the location of an initial process in response to detecting the event;

extracting an initial data address from one of the memory sources to determine the location of initial data to be used in the initial process in response to detecting the event;

executing executable code of the initial process;

Attorney Docket No.: 65,270-005

retrieving the initial data from one of the memory sources at the initial data address;

continuing execution of executable code of the initial process with the retrieved initial data to define an initial processed data set;

extracting an initial processed data address from one of the memory sources;

writing the initial processed data set to the initial processed data address;

extracting a second process address from one of the memory sources to determine the location of a second process to execute prior to the completion of the execution of the executable code of the initial process;

extracting a second data address from one of the memory sources to determine the location of second data to use in the second process;

executing executable code of the second process;

retrieving the second data from one of the memory source at the second data address;

continuing execution of executable code of the second process with the retrieved second data to define a second processed data set;

extracting a second processed data address from one of the memory sources;

writing the second processed data set to the second processed data address;

extracting a final process address from one of the memory sources to determine the location of a final process to execute;

executing executable code of the final process to halt communication of the system until the system detects the event.